Security Analysis of two Distance-Bounding Protocols

Mohammad Reza Sohizadeh Abyaneh*

Abstract. In this paper, we analyze the security of two recently proposed distance bounding protocols called the "Hitomi" and the "NUS" protocols. Our results show that the claimed security of both protocols has been overestimated. Namely, we show that the Hitomi protocol is susceptible to a full secret key disclosure attack which not only results in violating the privacy of the protocol but also can be exploited for further attacks such as impersonation, mafia fraud and terrorist fraud attacks. Our results also demonstrates that the probability of success in a distance fraud attack against the NUS protocol can be increased up to $(\frac{3}{4})^n$ and even slightly more, if the adversary is furnished with some computational capabilities.

Keywords: RFID, Privacy, Distance bounding protocol, Distance fraud

1 Introduction

Radio frequency identification (RFID) technology is widely being deployed today in many applications which require security, such as payment and access control applications. Although many solutions have been proposed to secure RFID systems, most of them are still susceptible to different attacks related to location such as: *distance fraud, mafia fraud* and *terrorist fraud* attacks. All of these attacks aim at suggesting a wrong assumption of the distance between a tag and a reader.

*Department of Informatics, University of Bergen

In distance fraud attack, a tag operates from out of the range where it is supposed to be. Mafia fraud attack, is a kind of man-in-the-middle attack in which a rogue tag circumvents the security mechanisms by getting right answers from the legitimate tag via a rogue reader, while both legitimate entities (legitimate reader and tag) remain unaware. In the terrorist attack, a legitimate tag colludes with the adversary, giving her the necessary information to access the system by impersonating it for a limited number of times.

The described attacks require simpler technical resources than tampering or cryptanalysis, and they cannot be prevented by ordinary security protocols that operate in the high layers of the protocol stack. The main countermeasure against these attacks is the use of *distance bounding* protocols, which verify not only that the tag knows the cryptographic secret, but also that is within a certain distance. To achieve this goal, distance bounding protocols must be tightly integrated into the physical layer [1].

In 1993, Brands and Chaum proposed the first distance bounding protocol [5]. Afterward, in 2005, Hancke and Kuhn [6] proposed the first distance-bounding protocol dedicated to RFID systems. This protocol has the drawback of giving the adversary this chance to succeed with the probability of $(\frac{3}{4})^n$ rather than $(\frac{1}{2})^n$ in distance and mafia fraud attacks, where n is a security parameter. Since then, there have been many solutions proposed either similar to Hancke and Kuhn [2, 7, 8, 10–12] or with different structures [5, 8, 9, 13–15]. However, they mostly have something in common; they all consist of three phases, the first and the last ones called *slow phases*, and the second one called the *fast phase*. The round trip time (RTT) of a bitwise challenge and response is measured n times during the fast phase to estimate the distance, while the slow phases include all the time-consuming operations.

Recently, two distance bounding protocols have been proposed by Lopez *et al* and Gürel *et al* called Hitomi [4] and Non-Uniform Stepping (NUS) [15] distance bounding protocols respectively. These protocols are claimed to provide privacy and resistance against distance, mafia and terrorist fraud attacks.

Our Contribution. In this paper, we apply a key disclosure attack to the Hitomi protocol and a distance fraud attack on the NUS protocol. Our analysis is framed in the formal framework introduced in [16].

Outline. The remainder of this paper is organized as follows. Section 2 includes a succinct description of the framework we do our security analysis within. In Sections 3, we describe the Hitomi protocol, its security claims and our key disclosure attack on it. In Section 4, we explain the NUS protocol and explain our distance fraud attack against it, and finally, Section 5 concludes the paper.

2 Preliminaries

See Section 5.2.3 of the thesis.

2.1 Notations

Here, we explain the notations used hereafter.

- x: Secret key of the tag.
- $f_x(.)$: Pseudo-Random Function operation with secret key x.
- $hw(.)$: Hamming Weight calculation function.
- N_R, N_T: Random numbers generated by the reader and the tag respectively.
- n: The length of registers considered as a security parameter.

2.2 Assumptions

The protocols described in this paper are executed under following assumptions:

- The tag and the reader share a long-term secret key x.
- Each tag has a unique identifier ID.
- The tag's capabilities supports a Pseudo-Random Function (f) and can perform bitwise operations.
- The reader and the tag agree on:
  - a security parameter n.
  - a public pseudo random function f with length of n bits.
  - a timing bound t_{max}
  - a fault tolerance threshold τ.

3 The Hitomi Protocol

3.1 Description

As stated in Section 1, being a distance bounding protocol, the Hitomi protocol (Figure 1) consists of three phases, two *slow phases* which are carried out at the first and final part of the protocol called *preparation phase* and *final phase* respectively. And the fast phase which is executed in between, called *rapid bit exchange phase*.

In the preparation phase, the reader chooses a random nonce (N_R) and transmits it to the tag. In return, the tag chooses three random numbers N_{T_1}, N_{T_2} and N_{T_3} and computes two temporary keys (k_1 and k_2) as (1) and (2).

$$k_1 = f_x(N_R, N_{T_1}, W) \tag{1}$$

$$k_2 = f_x(N_{T_2}, N_{T_3}, W') \tag{2}$$

where W and W' represent two constant parameters. By using these keys, the tag splits its permanent secret key x into two shares as *response registers*(i.e. $R^0 = k_1$ and $R^1 = k_2 \oplus x$). Finally, the tag transmits the 3-tuple $\{N_{T_1}, N_{T_2}, N_{T_3}\}$ to the reader.

The rapid bit exchange phase is a challenge and response phase with n rounds. In its i^{th} round, the reader generates a random challenge bit c_i and sends it to the tag while initializing a clock to zero. The tag receives c'_i which may not be equal to c_i due to errors or alterations in the channel. Immediately upon receiving c'_i, the tag responses with $r'_i = R_i^{c_i}$. The reader stops the clock after receiving r_i, which may not be equal to r'_i due to errors or alterations in the channel, and computes the round trip time (RTT) of this challenge and response transaction and stores it as Δt_i.

The final phase starts with computing and sending two following messages from the tag to the reader.

$$\begin{aligned} m &= \{c'_1, ..., c'_n, r'_1, ..., r'_n\} && (3) \\ t_B &= f_x(m, ID, N_R, N_{T_1}, N_{T_2}, N_{T_3}) && (4) \end{aligned}$$

Finally, the reader computes three kinds of errors and checks whether their summation is below a fault tolerance threshold as following.

- *errc*: the number of times that $c_i \neq c'_i$.
- *errr*: the number of times that $c_i = c'_i$ but $r_i \neq R_i^{c_i}$.

- *errt*: the number of times that $c_i = c'_i$ but the response delay Δt_i is more than a timing bound threshold $t_{max}(\Delta t_i > t_{max})$.

If the reader authentication is also demanded, the reader computes $t_A = f_x(N_R, k_2)$ and transmits it to the tag. Once the tag checks its correctness, the two entities are mutually authenticated.

The authors claim that the Hitomi protocol provides mutual authentication between the tag and the reader and also guarantees privacy protection. The authors argue that the success probability of the mafia and distance fraud attacks against their scheme is bounded by $(\frac{1}{2})^n$.

3.2 Key Disclosure Attack

In this section, we present an attacking scenario to the Hitomi protocol which leads to tag's secret key disclosure. Our main assumption in this attack is that the reader authentication is not demanded and so the protocol is executed without the optional message t_A. This allows an unauthorized reader(adversary) to query the tag several times without being detected.

Algorithm 1 portrays how an adversary is able to extract Δ bits of the tag's secret key by querying the tag m times.

The algorithm starts with the preparation phase in which at mth run, the adversary first generates a new random number N_R, sends it to the tag and receives the 3-tuple of $\{N_{T_1}, N_{T_2}, N_{T_3}\}$ in return.
The rapid bit exchange phase of the algorithm starts with generation of a challenge vector by the adversary which contains Δ bits of 1 and $n - \Delta$ bits of 0 ($c^{(m)}$). By sending the bits of this challenge vector to the tag in n rounds of the rapid bit exchange phase and receiving the responses, the adversary obtains $n - \Delta$ bits of $R^0 = k_1$ and Δ bits of $R^1 = k_2 \oplus x$.

We know that if the adversary is able to find k_1, she will be able to calculate k_2 by (2). Now, the adversary requires to search over all possible 2^Δ values for k_1. If we observe the output of $f_{k_1}(N_{T_2}^{(m)}, N_{T_3}^{(m)}, W')$ in the mth run of the protocol for 2^Δ times, each time with one different possible value of k_1, we will see that the number of values for the first Δ bits of k_2 ($k_{2(1)}, ..., k_{2(\Delta)}$) is less than 2^Δ. This can be calculated by a well-known problem in probability theory described in Remark 1.

Each k_2 nominates one $X_\Delta = (x_{(1)}, ..., x_{(\Delta)})$ for Δ bits of the tag's secret key (Line 16 of the Algorithm 1). So, each time the adversary queries the tag, she will obtain a set of potential candidates for X_Δ.

Reader		**Tag**
x, ID		x, ID
$N_R \in_R \{0,1\}^n$		
	$\xrightarrow{N_R}$	
		$N_{T_1}, N_{T_2}, N_{T_3} \in_R \{0,1\}^n$
	$\xleftarrow{N_{T_1}, N_{T_2}, N_{T_3}}$	
		$k_1 = f_x(N_R, N_{T_1}, W)$
		$k_2 = f_{k_1}(N_{T_2}, N_{T_3}, W')$
		$\begin{cases} R^0 = k_1 \\ R^1 = k_2 \oplus x \end{cases}$
	Start of rapid bit exchange	
	for $i = 1...n$	
$c_i \in \{0,1\}$		
start clock		
	$\xrightarrow{c_i}$	
		$r'_i = R_i^{c'_i}$
	$\xleftarrow{r_i}$	
stop clock		
Compute Δt_i		
	End of rapid bit exchange	
		$m = \{c'_1, ..., c'_n, r'_1, ..., r'_n\}$
		$t_B = f_x(m, ID, N_R, N_{T_1}, N_{T_2}, N_{T_3})$
	$\xleftarrow{m, t_B}$	
Check ID in the Database		
Compute R^0, R^1		
$err_c = \#\{i : c_i \neq c'_i\}$		
$err_r = \#\{i : c_i = c'_i, r_i \neq R_i^{c_i}\}$		
$err_t = \#\{i : c_i = c'_i, \Delta t_i > t_{max}\}$		
if $err_c + err_r + err_t \geq \tau$		
REJECT		
else		
$t_A = f_x(N_R, k_2)$		
	$\xrightarrow{t_A}$	
		Compute and Compare t_A

Fig. 1: *Hitomi Distance Bounding Protocols.*

If she continues querying the tag, each time she will obtain a set of different candidates.

These candidates can be removed from the list by further querying, unless they are nominated in the other runs. And the final candidate is the one which has been in the candidate list in all the queries. The number of times that the adversary must query the tag to be left with only one candidate is calculated by (7) and plotted in Figures 2 and 3.

Remark 1. Consider the process of tossing b balls into b bins. The tosses are uniformly at random and independent of each other. The

Algorithm 1 Δ bit secret key disclosure

Inputs: n, Δ, W, W'
Outputs: m, Δ bits of secret key x $(x_1, ..., x_\Delta)$

$m \leftarrow 1$ {number of required runs of the protocol}
repeat
  $NumberofCandidates \leftarrow 0$
  $FinalCandidate \leftarrow 0$
  $\{counter(1), ..., counter(2^\Delta)\} \leftarrow \{0x0, ..., 0x0\}$
  $\{CandidateFlag(1), ..., CandidateFlag(2^\Delta)\} \leftarrow \{0x0, ..., 0x0\}$
  Generate $N_R^{(m)}$ and Send to the tag.
  Receive $N_{T_1}^{(m)}, N_{T_2}^{(m)}, N_{T_3}^{(m)}$
  $c^{(m)} \leftarrow (\underbrace{1, ..., 1}_{\Delta}, \underbrace{0, ..., 0}_{n-\Delta})$
  send the challenges to the tag in n rounds and receive the responses.
  $r^{(m)} \leftarrow (r_{(1)}^{(m)}, ..., r_{(n)}^{(m)})$
  $(k_{1(\Delta+1)}, ..., k_{1(n)}) \leftarrow (r_{(\Delta+1)}^{(m)}, ..., r_{(n)}^{(m)})$
  for $i = 0$ **to** $2^\Delta - 1$ **do**
    $(k_{1(1)}, ..., k_{1(\Delta)}) \leftarrow Decimal2Binary(i)^*$
    $(k_{2(1)}, ..., k_{2(n)}) \leftarrow f_{k_1}(N_{T_2}^{(m)}, N_{T_3}^{(m)}, W')$
    $(x_{(1)}, ..., x_{(\Delta)}) \leftarrow (k_{2(1)}, ..., k_{2(\Delta)}) \oplus (r_{(1)}^{(m)}, ..., r_{(\Delta)}^{(m)})$
    $l \leftarrow Binary2Decimal(x_{(1)}, ..., x_{(\Delta)})^{**}$
    if $CandidateFlag(l) = 0$ **then**
      $counter(l) \leftarrow counter(l) + 1$
      $CandidateFlag(l) \leftarrow 1$
    end if
  end for
  for $j = 1$ **to** 2^Δ **do**
    if $counter(j) = m$ **then**
      $NumberofCandidates \leftarrow NumberofCandidates + 1$
      $FinalCandidate \leftarrow j$
    end if
  end for
  $m \leftarrow m + 1$
until $NumberofCandidates = 1$
$(x_{(1)}, ..., x_{(\Delta)}) \leftarrow Decimal2Binary(FinalCandidate)$
return $m, (x_{(1)}, ..., x_{(\Delta)})$

* Decimal2Binary(.) outputs the binary representation of a given decimal number.
** Binary2Decimal(.) outputs the decimal representation of a given binary number.

probability of not falling any ball into a particular bin can be calculated by (5) [17].

$$\text{Pr(one particular bin remains empty)} = p_0 = \left(1 - \frac{1}{b}\right)^b \approx \frac{1}{e},\ b \gg 1 \tag{5}$$

Hence, the probability that a ball does not remain empty is simply $p_1 = 1 - p_0$. Due to independency, if we repeat the same experiment for m trials, the probability that one particular bin remains empty at least in one of m trials is $1 - p_1^m$. Now, we can calculate the probability that all bins experience to be empty at least in one of m trials ($Pr(Success)$) by (6).

$$\begin{aligned} Pr(Success) &= (1 - p_1^m)^b = \left(1 - \left(1 - \left(1 - \frac{1}{b}\right)^b\right)^m\right)^b \\ &\approx \left[1 - \left(1 - \frac{1}{e}\right)^m\right]^b,\ b \gg 1 \end{aligned} \tag{6}$$

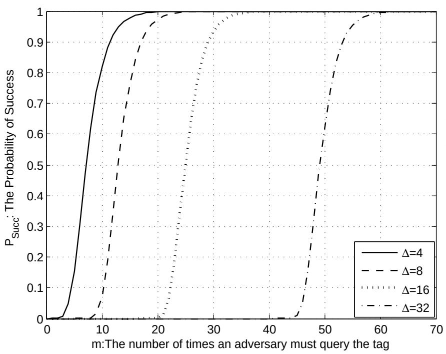
For our problem it is only required to substitute b with 2^Δ and we will have:

$$P_{Succ} = Pr(Success) = \left(1 - \left(1 - \left(1 - \frac{1}{2^\Delta}\right)^{2^\Delta}\right)^m\right)^{2^\Delta} \tag{7}$$

Figure 2 illustrates the probability of success calculated in (7) while the number of protocol runs are increased. The figures have been plotted for $\Delta = 4,8,16$ and 32, which should be chosen according to computational constraints. So far, we have accomplished to find the first Δ bits of the tag's secret key with a certain probability. In a similar vein, one can find other bits of the secret key by choosing a different challenge vector (e.g. for finding $(x_{(\Delta+1)}, ..., x_{(2\Delta)})$ the challenge should be chosen like (8) and the above algorithm should be executed another time).

$$c = (\underbrace{0, ..., 0}_{\Delta}, \underbrace{1, ..., 1}_{\Delta}, \underbrace{0, ..., 0}_{n-2\Delta}) \tag{8}$$

In this way, the adversary accomplishes to find the whole tag's secret key, if she can query the tag for enough times. Figure 3 illustrates the number of runs of the protocol which an adversary must query the tag and its probability of success to find the entirety of tags's secret



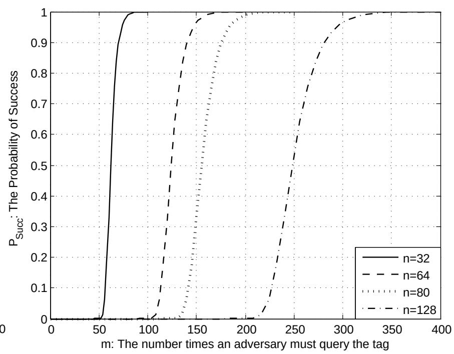


Fig. 2: *Adversary success probability to find Δ bits of the secret key.*

Fig. 3: *Adversary success probability to find the whole secret key for $\Delta = 16$.*

key, assuming that her computational capability is limited to $2^{\Delta} = 2^{16}$ computations. The computations include: searching over 2^{Δ} values of k_1, finding k_2 for each k_1 and candidate one X_{Δ}.
The graphs have been plotted for four different key sizes $n = 32$, 64, 80 and 128. For instance, the adversary is required to query the tag about 70, 140, 175 and 280 times to find the tag's secret keys of size 32, 64, 80 and 128 bits with the probability of about 0.9 respectively.

It is obvious that having this attack accomplished, the adversary is able to easily either track or impersonate the tag in further interrogations. The information elicited in this attack also paves the way for performing other attacks such as mafia or terrorist fraud attacks.

4 The NUS Protocol

4.1 Description

The NUS protocol (Figure 4) also consists of three phases, two *slow phases* a fast called *rapid bit exchange phase*.
In the first slow phase, the reader chooses a random nonce(N_R) and transmits it to the tag. In return, the tag chooses another random number(N_T) and computes the response register $R = f_x(N_R, N_T)$, which is of length $2n$. The tag then initializes the variables $j1, j2, k1$ and $k2$ to $1, n, 0$ and $2n + 1$ respectively and sends back N_T to the reader.

In the i^{th} round of the rapid bit exchange phase, the reader generates a random challenge bit c_i and sends it to the tag while initializing a clock to zero. The tag receives c'_i which may not be equal to c_i due to errors or alterations in the channel. Immediately upon receiving c'_i, the tag sends the bit r'_i, computed according to the procedure shown in Figure 4.

The final phase concludes with sending the message m which consists of all challenges the tag has received, from the tag to the reader and finally, the error computation which is almost the same as in the Hitomi protocol.

The authors claim that the success probability of the distance, mafia and terrorist fraud attacks against the NUS protocol is bounded by $(\frac{1}{2})^n$.

4.2 DISTANCE FRAUD ATTACK

In this section, we present a distance fraud attack on the NUS protocol in two different forms in white-box model: *restricted adversary* and *powerful adversary*. The main assumption we have is that the adversary is located at zone Z_1, i.e. at the i^{th} round of the rapid bit exchange phase, the adversary accesses to the value of the challenge bit in previous round c_{i-1}, before generating current response r_i. This assumption implies that the adversary is able to update the registers $j1, j2, k1$ and $k2$ and she is aware of their correct current values, before she generates the response.

Restricted adversary

The adversary is allowed to run only once the pseudo-random function f function to compute R and observe its content before any response. The probability of success for the distance fraud attack in this model can be calculated by (9).

$$\begin{aligned} P_{dis} &= Pr(\text{Success}|x_{j1}x_{j2} = 00)Pr(x_{j1}x_{j2} = 00) \\ &+ Pr(\text{Success}|x_{j1}x_{j2} = 01)Pr(x_{j1}x_{j2} = 01) \\ &+ Pr(\text{Success}|x_{j1}x_{j2} = 10)Pr(x_{j1}x_{j2} = 10) \\ &+ Pr(\text{Success}|x_{j1}x_{j2} = 11)Pr(x_{j1}x_{j2} = 11) \end{aligned} \tag{9}$$

If $x_{j1}x_{j2} = 00$ and without knowing c_i, the adversary should anticipate the right response(r_i) between R_{k1+1} and R_{k2-1}. Let us define the probability of equality of these two bits by (10).

Reader		**Tag**
x, ID		x, ID
$N_R \in_R \{0,1\}^n$		
	$\xrightarrow{N_R}$	
		$N_T \in_R \{0,1\}^n$
		$R = f_x(N_R, N_T), \lvert R \rvert = 2n$
		$j1 = 1, j2 = n$
		$k1 = 0, k2 = 2n+1$
	$\xleftarrow{N_T}$	
	Start of rapid bit exchange	
	for $i = 1...n$	
$c_i \in \{0,1\}$		
start clock		
	$\xrightarrow{c_i}$	
		if $c'_i = 0$
		if $x_{j1} = 0$
		$k1 = k1+1,\ r'_i = R_{k1}$
		else
		$k2 = k2-2,\ r'_i = R_{k2}$
		$j1 = j1+1$
		else
		if $x_{j2} = 0$
		$k2 = k2-1,\ r'_i = R_{k2}$
		else
		$k1 = k1+2,\ r'_i = R_{k1}$
		$j2 = j2-1$
	$\xleftarrow{r_i}$	
stop clock, Compute Δt_i		
	End of rapid bit exchange	
		$m = \{c'_1, ..., c'_n\}$
	$\xleftarrow{m}$	
$err_c = \#\{i : c_i \neq c'_i\}$		
Compute $(r'_i)'s$ using $(c'_i)'$s		
$err_r = \#\{i : c_i = c'_i, r_i \neq r''_i\}$		
$err_t = \#\{i : c_i = c'_i, \Delta t_i > t_{max}\}$		
Checks $err_c + err_r + err_t \leq \tau$		

Fig. 4: *The NUS Distance Bounding Protocol*

$$P_{eq} = \Pr(R_{k1+1} = R_{k2-1}) \tag{10}$$

So, we have,

$$\begin{aligned} Pr(\text{Success}|x_{j1}x_{j2} = 00) &= Pr(\text{Success}|x_{j1}x_{j2} = 00, R_{k1+1} = R_{k2-1})(P_{eq}) \\ &+ Pr(\text{Success}|x_{j1}x_{j2} = 00, R_{k1+1} \neq R_{k2-1})(1 - P_{eq}) \end{aligned}$$

If $R_{k1+1} = R_{k2-1}$, the adversary can simply outputs either of these two bits and succeeds with the probability 1. Otherwise, she outputs a random bit and she will have the success probability of $\frac{1}{2}$. So,

$$Pr(\text{Success}|x_{j1}x_{j2} = 00) = 1 \times P_{eq} + \frac{1}{2} \times (1 - P_{eq}) = \frac{(1 + P_{eq})}{2} \quad (11)$$

We can do similar calculations for other three possibilities of $x_{j1}x_{j2}$. Since all four possibilities of $x_{j1}x_{j2}$ are equally likely, we have the probability of success for a distance fraud attack in one round as (12).

$$P_{dis} = \frac{(1 + P_{eq})}{2} \quad (12)$$

In a similar vein, one can show that due to independency of the n rounds, the adversary obtains the success probability of $(\frac{1+P_{eq}}{2})^n$ for n rounds. If we assume that zeros and ones are equally likely, P_{eq} equals to $\frac{1}{2}$ and for n rounds we have:

$$P_{dis} = (\frac{3}{4})^n \quad (13)$$

Powerful adversary

Our main assumptions in this attack are as following. We assume that, there is a 1-second latency between the preparation and rapid bit exchange phases of the protocol. It implies that the adversary can run the pseudo-random function f for c times between the preparation and the rapid bit exchange phases, where c the number of a simple random number function like a hash function that can be computed per second on a single PC [16].

In [16], Avoine *et al* has presented an instance of a distance fraud attack against a white-box-modeled tag in Hancke and Kuhn protocol. They have devoted the white-box modeled tag's capabilities to minimize the hamming weight difference of n-bit response registers in the Hancke and Kuhn protocol($hw(R^0 \oplus R^1)$). They have proved that if $P_i = Pr(success|(hw(R^0 \oplus R^1) = i))$, the probability of success in the distance fraud attack can be calculated by (14).

$$P_{dis} = \left(\frac{1}{2}\right)^{cn} \times \left(\sum_{i=0}^{i=n-1} (P_i) \left[\left(\sum_{j=i}^{j=n} \binom{n}{j} \right)^c - \left(\sum_{j=i+1}^{j=n} \binom{n}{j} \right)^c \right] + 1 \right) \quad (14)$$

	n=32	n=64	n=80	n=128
Claimed Security	2.3283E-10	5.4210E-20	8.2718E-25	2.9387E-39
Restricted Adversary	1.0045E-4	1.0090E-8	1.0113E-10	1.0183E-16
Powerful Adversary	0.0035	4.5101E-7	4.7459E-9	5.1498E-15

Table 1: *Comparison of the probability of success for distance fraud attack against the NUS protocol for $c = 2^{23} \approx E6$.*

In order to utilize (14) for our purpose, we define $P_i = Pr(Success|hw(R) = i)$. This implies that, we devote the tag's capability to minimize the hamming weight of the response register R in the NUS protocol. Having this in mind and by using (12), we can calculate P_i for n rounds as following.

$$\begin{aligned} P_{eq} &= (\frac{i}{2n})^2 + (\frac{2n-i}{2n})^2 = 1 + \frac{i^2 - 2in}{2n^2} \\ P_i &= P_{dis} = \left[\frac{(1+P_{eq})}{2}\right]^n = \left(1 + \frac{i^2 - 2in}{4n^2}\right)^n \end{aligned} \tag{15}$$

As the response register R in the NUS protocol is of length $2n$, we only need to substitute n by $2n$ and P_i by (15) in (14). Table 1 compares the claimed security of the NUS protocol and our results in restricted and powerful adversary models in terms of the probability of success of an adversary in the distance fraud attack. For example, for $n = 32$, the probability of success in the distance fraud attack in a restricted adversary model is 1.0045E-4. This probability improves to 0.0035 in a powerful adversary model for $c = 2^{23}$ which roughly represents the number of hashes that can be computed today per second on a single PC [16]. These probabilities are remarkably beyond the claimed security $(\frac{1}{2})^{32} = 2.3283\text{E-}10$.

5 Conclusions

The design of a secure distance bounding protocol which can resist against the existing attacks for RFID systems is still challenging. Despite of interesting proposals in the literature, this field still lacks a concrete solution.

Recently, two solutions have been proposed for this purpose called the Hitomi and the NUS distance bounding protocols. We presented a secret key disclosure attack on the former and a distance fraud attack on the latter protocol. Our results showed that the security margins which was expected to be yielded by them have been overestimated.

We showed that the Hitomi protocol is vulnerable to a full secret key disclosure attack by querying the tag several times. In addition, the probability of success in a distance fraud attack against the NUS protocol was shown to be able to be increased up to $(\frac{3}{4})^n$, if the adversary gets close enough to the reader. This probability can even be slightly improved, if the tag has some computational capabilities.

References

[1] Jorge Munilla Fajarado, A. Peinado Dominguez. Security in RFID and Sensor Networks. First Edition, ISBN:978-1-4200-6839-9, Auerbach publication, 2009.

[2] Gildas Avoine and Aslan Tchamkerten. An efficient distance bounding RFID authentication protocol: balancing false-acceptance rate and memory requirement. *In Information Security Conference - ISC'09*, volume 5735 of Lecture Notes in Computer Science, Pisa, Italy, September 2009.

[3] Orhun Kara, Süleyman Karda, Muhammed Ali Bingöl, Gildas Avoine: Optimal Security Limits of RFID Distance Bounding Protocols editors *Radio Frequency Identification: Security and Privacy Issues - RFIDSec10*, volume 6370 of Lecture Notes in Computer Science, pages 220–238, Istanbul, Turkey, June 2010.

[4] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and J. C. A. van der Lubbe. Shedding Some Light on RFID Distance Bounding Protocols and Terrorist Attacks. arXiv.org, Computer Science, Cryptography and Security, June , 2010.

[5] Stefan Brands and David Chaum. Distance-bounding protocols. *In EUROCRYPT'93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pages 344–359, Secaucus, NJ, USA, 1994.

[6] Gerhard Hancke and Markus Kuhn. An RFID Distance Bounding Protocol. *In Conference on Security and Privacy for Emerging Areas in Communication Networks - SecureComm 2005*, pages 67–73, Athens, Greece, September 2005.

[7] Chong Hee Kim and Gildas Avoine. RFID Distance Bounding Protocol with Mixed Challenges to Prevent Relay Attacks. *In 8th International Conference on Cryptology And Network Security - CANS'09*, Japan, December 2009.

[8] Chong Hee Kim, Gildas Avoine, Francois Koeune, Francois-Xavier Standaert, and Olivier Pereira. The Swiss-Knife RFID Distance Bounding Protocol. *In P.J. Lee and J.H. Cheon, editors, International Conference on Information Security and Cryptology - ICISC, volume 5461 of Lecture Notes in Computer Science*, pages 98–115, Seoul, Korea, December 2008.

[9] Jorge Munilla, Andres Ortiz, and Alberto Peinado. Distance Bounding Protocols with Void-Challenges for RFID. *In Workshop on RFID Security - RFIDSec'06*, Graz, Austria, July 2006.

[10] Jorge Munilla and Alberto Peinado. Security Analysis of Tu and Piramuthu's Protocol. *In New Technologies, Mobility and Security - NTMS'08*, pages 1–5, Tangier, Morocco, November 2008.

[11] Jason Reid, Juan Gonzalez Neito, Tee Tang, and Bouchra Senadji. Detecting relay attacks with timing based protocols. *In Feng Bao and Steven Miller, editors, Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security - ASIACCS '07*, pages 204–213, Singapore, Republic of Singapore, March 2007.

[12] Yu-Ju Tu and Selwyn Piramuthu. RFID Distance Bounding Protocols. In First International EURASIP Workshop on RFID Technology, Vienna, Austria, September 2007.

[13] Rolando Trujillo-Rasua, Benjamin Martin, and Gildas Avoine. The Poulidor Distance-Bounding Protocol. *In Radio Frequency Identification: Security and Privacy Issues - RFIDSec10, volume 6370 of Lecture Notes in Computer Science*, page 239–257, Istanbul, Turkey, June 2010.

[14] Pedro Peris-Lopez and Julio C. Hernandez-Castro and Juan M. E. Tapiador and Esther Palomar and Jan C.A. van der Lubbe. Cryptographic Puzzles and Distance-bounding Protocols: Practical Tools for RFID Security. *In IEEE International Conference on RFID*, Orlando,

2010.

[15] Ali Özhan Gürel, Atakan Arslan, Mete Akgün . Non-Uniform Stepping Approach to RFID Distance Bounding Problem. *Fifth International Workshop on Data Privacy Management - DPM'10*, volume 6370 of Lecture Notes in Computer Science, Athens, Greece, September 2010.

[16] G. Avoine, M. A. Bingol, S. Kardas, C. Lauradoux, and B. Martin. A Formal Framework for Cryptanalyzing RFID Distance Bounding Protocols. Cryptology ePrint Archive, Report 2009/543, 2009.

[17] W. Feller, An Introduction to Probability Theory and its Applications, Wiley India Pvt. Ltd., 2008.

[18] D. Dolev and A. C.-C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory, 29(2)*:pages 198–207, 1983.